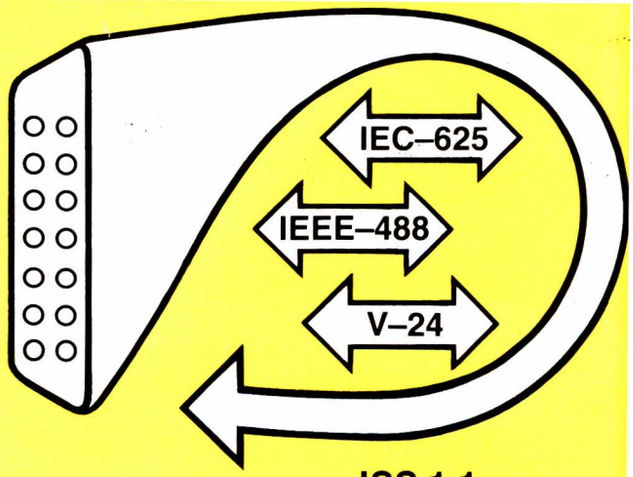
T&M

Instrumentation System Support



ISS 1.1

CBM COMMODORE (MODELS 2001–8, –16, –32, 3016 and 3032) AS INSTRUMENTATION CONTROLLER.

Using the CBM Commodore as instrumentation controller, a number of points should be observed. Realize that this note does not have the intention to replace the CBM Commodore Users Manual, but only will give some hints to build up a controller program.

1. GENERAL

When using the CBM Commodore as IEC-bus controller, attention should be given to the following points:

Controller:

The CBM must be the **only** controller in a system.

Message delimiter:

The CBM has no facility to change the message delimiter. It is fixed at 'CR', contrary to the 'ETX' code as laid down by IEC 625-2.
If the connected device does not have 'CR' as delimiter and no facility to change to it, it should sent the END message to indicate that the data is completed. (EOI line)

REN (remote enable):

The REN line is connected to ground (earth), consequently the REN message is continuously 'true'.

IFC (interface clear):

After POWER ON, the CBM automatically sends the IFC message for a period of 100 ms.
Note that IFC is not implemented in CBM BASIC.

The Handshake Procedure:

When the PEEK (memory reading) and POKE (memory writing) statements are used to send or receive data, the handshake procedure must also be performed with these PEEK and POKE instructions.
(See Section 2.2.1.)

SRQ (service request):

Using BASIC commands, the CBM is not capable of recognising a service request.
Also the serial poll routine is not implemented in CBM BASIC.
However, the above-mentioned functions can be programmed with the PEEK and POKE commands of the CBM
(see Section 2.2.3)

PHILIPS

PHILIPS

Addressed and Universal Commands:

The CBM cannot implement the commands from the addressed and universal command group (columns 0 and 1 of the ISO 7-bit coding table) in BASIC.
These commands can be handled by using special routines with PEEK and POKE commands. For program examples, see Section 2.2.2.).

Input data preceeded by spaces:

When the first byte(s) of input data should be a space, these spaces are ignored by the CBM when data is taken in by an INPUT statement.

Time out:

The following limitations should be noted to avoid any loss of data:

- when the CBM is a listener, it requires DAV to go low within 65 ms.
- when the CBM is a talker it requires NDAC to go high within 65 ms after it has set DAV low.

The CBM ceases to transfer if these requirements are exceed and sets the appropriate status word (ST).

Survey of Status Words:

Status Bit Position	Status Numeric Value	IEC Read/Write Status
0	1	Time out on write
1	2	Time out on read
2	4	
3	8	
4	16	
5	32	
6	64	EOI line
7	128	Device not present

Basic Statements and Parameters for IEC-bus Handling:

	example
OPEN <f>,<d>,<s>,<v>	OPEN 3, 7, 2, name.
INPUT #<f>,	INPUT # 3, A
GET # <f>,	GET # 3, B
PRINT # <f>,	PRINT # 3, C
CMD <f>	CMD 3
CLOSE <f>	CLOSE 3

<f> =	logical file number, an integer between 0 and 255
<d> =	device address, an integer between 0 and 30
<s> =	secondary command, an integer between 0 and 30
<v> =	variable, specifies a data item or file name

NOTES:

The device addresses 0 – 3 and the secondary command 0 – 6 are used by the CBM itself and are not available for the IEC-bus devices.

The secondary command and the variable are optional facilities.

2. PROGRAMMING HINTS

2.1. General

This section gives program examples and hints for requirements that are not implemented in CBM BASIC. To enable these facilities it is necessary to provide access from the I/O memory of the CBM to the IEC-bus.

The following table lists the I/O memory addresses assigned to the IEC-bus.

CBM HARDWARE ADDRESSES FOR IEC-BUS LINES								
BIT / MEMORY	7 (128)	6 (64)	5 (32)	4 (16)	3 (8)	2 (4)	1 (2)	0 (1)
59424	◄ DIO-8 (IN) ►							
59425	ATN (IN)				NDAC (OUT)			
59426	◄ DIO-8 (OUT) ►							
59427	SRQ (IN)				DAV (OUT)			
59456	DAV (IN)	NRFD (IN)				ATN (OUT)	NRFD (OUT)	NDAC (IN)
59408		EOI (IN)						
59409					EOI (OUT)			

Using the PEEK and POKE commands of the CBM it is possible to read or write these memory addresses by setting or resetting bits. Furthermore, testing can be performed on a particular bit in a memory address. These actions can be programmed as follows:

Testing a bit:

B = PEEK (An) AND { 1 / 2 / 4 / 128 }

If B= { 0 / 1 } GOTO

Resetting a bit:

POKE An, PEEK (An) AND {
(255-1) – set bit 0 to zero
(255-2) – set bit 1 to zero
(255-4) – set bit 2 to zero
(255-128) – set bit 7 to zero }

Setting a bit:

POKE An, PEEK (An) OR {
1 – set bit 0 to one
2 – set bit 1 to one
4 – set bit 2 to one
128 – set bit 7 to one }

NOTE: An = memory location.

In the program examples listed, these fundamental program lines are used.
To keep the program lines as short as possible, it is recommended that the memory addresses are first labelled.

e.g. 100 A1 = 59424

The memory location 59424 can then be used in the program as A1.

2.2. PROGRAM EXAMPLES

In the following program examples a number of subroutines with additional comment are given. These subroutines are general purpose routines and can be used in a main program directly. In Section 2.2.6. a main program for the PM2528 using these subroutines is given. In all program examples the memory addresses are labelled as follows:

```
170 A1=59426:                              REM A1-5 ARE LABELS OF MEMORY LOCATIONS
180 A2=59427
190 A3=59456
200 A4=59425
210 A5=59424
```

As device address, 22 is used.
For other device addresses, the listener and talker address can be calculated as follows:
Listener address (decimal value) = Device address + 32
Talker address (decimal value) = Device address + 64.

2.2.1. Handshake procedure.

```
640 REM*************************************************
650 REM        SUBR. SOURCE HANDSHAKE                  *
660 REM*************************************************
670 ::
680 POKEA4,PEEK(A4)OR8:                    REM SET DAC=1
690 POKEA3,PEEK(A3)OR67:                   REM SET RFD=1;SET DAC AND RFD INPUTS =1
700 T=PEEK(A3)AND64:IFT=0THEN700:          REM WAIT UNTIL RFD=1
710 POKEA2,PEEK(A2)AND(255-8):             REM SET DAV=1
720 T=PEEK(A3)AND1:IFT=0THEN720:           REM WAIT UNTIL DAC=1
730 POKEA2,PEEK(A2)OR8:                    REM SET DAV=0
740 POKEA1,255:                            REM ENABLE DIO1-8
750 POKEA4,PEEK(A4)AND(255-8):             REM SET DAC=0
760 RETURN
770 ::
780 ::
790 REM*************************************************
800 REM        SUBR. ACCEPTOR HANDSHAKE                *
810 REM*************************************************
820 ::
830 POKEA5,255:                            REM CLEAR DATA INPUT
840 POKEA4,PEEK(A4)AND(255-8):             REM SET DAC=0
850 POKEA3,PEEK(A3)OR2:                    REM SET RFD=1
860 POKEA3,PEEK(A3)OR4:                    REM SET ATN=0
870 T=PEEK(A3)AND(255-128):IFT=1THEN870:REM WAIT UNTIL DAV=1
880 POKEA3,PEEK(A3)AND(255-2):             REM SET RFD=0
890 B=(255-PEEK(A5)):                      REM B=INPUT DATA
900 POKEA4,PEEK(A4)OR8:                    REM SET DAC=1
910 T=PEEK(A3)OR128:IFT=0THEN910:          REM WAIT UNTIL DAV=0
920 POKEA4,PEEK(A4)AND(255-8):             REM SET DAC=0
930 POKEA3,PEEK(A3)OR2:                    REM SET RFD=1
940 RETURN
```

2.2.2. Send commands or addresses.

Universal commands, addressed commands and addresses have to be sent with ATN =1
Therefore the next sequence should be followed:

- Set the DIO output memory location
- Set ATN = 1
- Call Subr. Source handshake
- Set ATN = 0

For an addressed command, first the listen address of the device should be sent.

Example:

Addressed command Group Execute Trigger (GET).

```
360 POKEA1,(255-54):GOSUB550:          REM SEND LISTEN ADDRESS (DEC. VALUE)
370 POKEA1,(255-8):GOSUB550:           REM SEND GET (GROUP EXECUTE TRIGGER)
520 ::
530 ::
540 REM*************************************************************************
550 REM     SUBR. SEND COMMAND OR ADDRESS                                      *
560 REM*************************************************************************
570 ::
580 POKEA3,PEEK(A3)AND(255-4):         REM SET ATN=1
590 GOSUB650:                          REM SUBR. SOURCE HANDSHAKE
600 POKEA3,PEEK(A3)OR4:                REM SET ATN=0
610 RETURN
READY.
```

NOTE: The addressed command GET is very often used as START command of a device e.g. in the PM2527.

2.2.3. Service request and serial poll.

In this program example, the CBM waits until a device has set the SRQ line.
In a complete system it is more pratical to look at the SRQ line regularly and to continue the main program if the SRQ line is not set. Note that the BASIC device address 22 is not used, but the real decimal value of the talker and listener address.

Program example:

```
260 T=PEEK(A2)AND128:IFT=0THEN260:     REM WAIT UNTIL SERVICE REQUEST (SRQ)
270 D=86:GOSUB410:                     REM SERIAL POLL DEV. ADDR. 22 (=86DEC)
380 ::
390 ::
400 REM*************************************************************************
410 REM          SUBROUTINE  SERIAL POLL                                       *
420 REM*************************************************************************
430 ::
440 POKEA1,(255-24):GOSUB550:          REM SEND SPE (SERIAL POLL ENABLE)
450 POKEA1,(255-D):                    REM DIO1-8=TALK ADDRESS (D=DEC. VALUE)
460 POKEA3,PEEK(A3)AND(255-4):         REM SET ATN=1
470 GOSUB650:                          REM SEND TALKADDRESS D
480 GOSUB800:                          REM SUBR. ACCEPTOR HANDSHAKE (READ ST)
490 POKEA1,(255-95):GOSUB550:          REM SEND UNT (UNTALK)
500 POKEA1,(255-25):GOSUB550:          REM SEND SPD (SERIAL POLL DISABLE)
510 RETURN
```

2.2.4. Testing of the status byte.

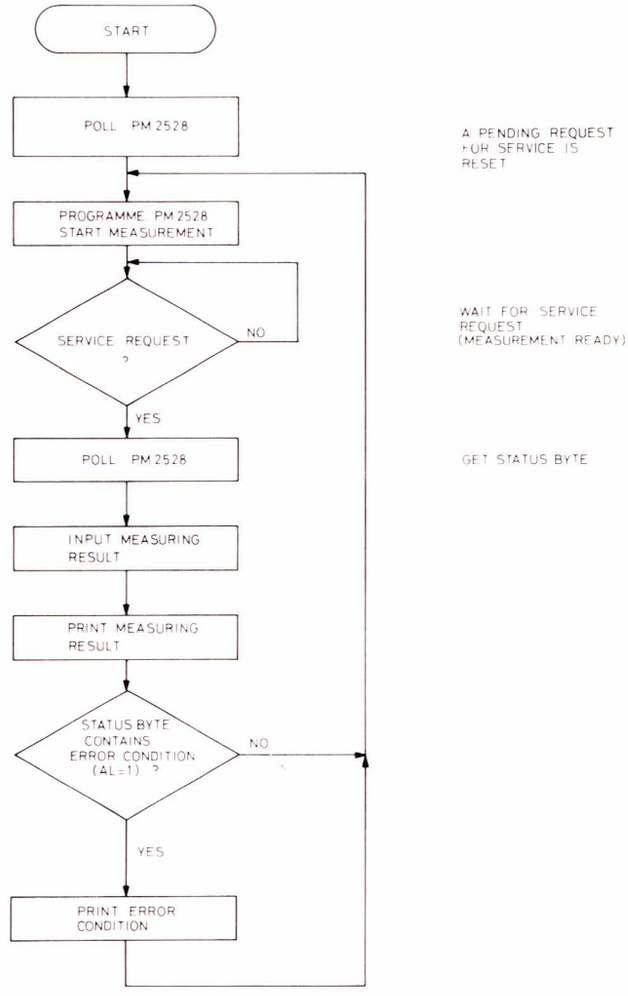
Next program example gives the test on overload and/or crest factor exceeded of the PM2528.
Therefore first a test on the alarm bit (bit 6 = 1= decimal value 32)is excecuted.

```
310 T=B AND 32:IFT=32THENGOSUB980:       REM IF AL-BIT=1 THEN READ ERROR (=C$)
950 ::
960 ::
970 REM*************************************************************************
980 REM      SUBR. READ ERROR                                                   *
990 REM*************************************************************************
1000 ::
1010 T=B AND 15:                         REM MASK BIT 4-7 OF STATUS BYTE
1020 IFT=1THENC$="OL":                   REM OVERLOAD
1030 IF T=2THENC$="CF":                  REM CRESTFACTOR EXCEEDED
1040 IF T=3THENC$="OL+CF":               REM OVERLOAD+CRESTFACTOR EXCEEDED
1050 RETURN
```

2.2.5. Program example to control the PM2528 with the CBM Commodore.

In this program a number of the above-mentioned subroutines are used.

Flow chart:



```
100 GOTO170
110 ********************************************************************************
120 THIS IS A PROGRAM FOR PM2528 CONNECTED TO CBM VIA IEC-BUS                      *
130 IT PROGRAMMES AND TRIGGERS PM2528,WAITS FOR SERVICE REQUEST (MEASUREMENT      *
140 DATA AVAILABLE) ,FETCHES STATUS BYTE (SERIAL POLL),ANALYZES STATUS BYTE       *
150 AND PRINTS MEASUREMENT DATA AND ERROR IN ALARM CONDITION                      *
155 SETTINGS OF PM2528 ON REAR:                                                   *
156 .    ADDRESS =10110 (A5 A4 A3 A2 A1)     0=OFF(PRESSED)  1=ON(PRESSED)       *
157 .    TON     =0                                                               *
158 .    SRQ OFF =0                                                               *
160 ********************************************************************************
170 A1=59426:  57090                    REM A1-5 ARE LABELS OF MEMORY LOCATIONS
180 A2=59427   57091
190 A3=59456   57120
200 A4=59425   57089
210 A5=59424   57088
220 OPEN3,22:                           REM DEVICE 22=ADDR. 10110
230 D=86:GOSUB410:                      REM SERIAL POLL DEV.  ADDR. 22 (=86DEC)
240 PRINT#3,"F00,R0,T2,D1,E1":          REM PROGRAMME VDC,AUTO-RANGE,
250 REM                                     EXT.START MODE,DATA READY REQUEST,
255 REM                                     START MEASUREMENT
260 T=PEEK(A2)AND128:IFT=0THEN260:      REM WAIT UNTIL SRQ(IN)
270 D=86:GOSUB410:                      REM SERIAL POLL DEV. ADDR.22 (=86DEC)
280 INPUT#3,A$:                         REM GET MEASUREMENT DATA
290 PRINTA$;:                           REM DISPLAY MEASUREMENT DATA
300 C$=""
310 T=BAND32:IFT=32THENGOSUB980:        REM IF AL-BIT=1 THEN READ ERROR (=C$)
320 PRINT"  VOLT   ";C$
330 GOTO240
340 CLOSE3
350 END
380 ::
390 ::
400 REM*********************************************************************************
410 REM     SUBROUTINE  SERIAL POLL                                                    *
420 REM*********************************************************************************
430 ::
440 POKEA1,(255-24):GOSUB550:           REM SEND SPE (SERIAL POLL ENABLE)
450 POKEA1,(255-D):                     REM DIO1-8=TALK ADDRESS (D=DEC. VALUE)
460 POKEA3,PEEK(A3)AND(255-4):          REM SET ATN=1
470 GOSUB650:                           REM SEND TALKADDRESS D
480 GOSUB800:                           REM SUBR. ACCEPTOR HANDSHAKE (READ ST)
490 POKEA1,(255-95):GOSUB550:           REM SEND UNT (UNTALK)
500 POKEA1,(255-25):GOSUB550:           REM SEND SPD (SERIAL POLL DISABLE)
510 RETURN
520 ::
530 ::
540 REM*********************************************************************************
550 REM     SUBR. SEND COMMAND OR ADDRESS                                              *
560 REM*********************************************************************************
570 ::
580 POKEA3,PEEK(A3)AND(255-4):          REM SET ATN=1
590 GOSUB650:                           REM SUBR. SOURCE HANDSHAKE
600 POKEA3,PEEK(A3)OR4:                 REM SET ATN=0
610 RETURN
620 ::
630 ::
```

```
640 REM*****************************************************************************
650 REM     SUBR. SOURCE HANDSHAKE                                                 *
660 REM*****************************************************************************
670 ::
680 POKEA4,PEEK(A4)OR8:                    REM SET DAC=1
690 POKEA3,PEEK(A3)OR67:                   REM SET RFD=1;SET DAC AND RFD INPUTS =1
700 T=PEEK(A3)AND64:IFT=0THEN700:          REM WAIT UNTIL RFD=1
710 POKEA2,PEEK(A2)AND(255-8):             REM SET DAV=1
720 T=PEEK(A3)AND1:IFT=0THEN720:           REM WAIT UNTIL DAC=1
730 POKEA2,PEEK(A2)OR8:                    REM SET DAV=0
740 POKEA1,255:                            REM ENABLE DIO1-8
750 POKEA4,PEEK(A4)AND(255-8):             REM SET DAC=0
760 RETURN
770 ::
780 ::
790 REM*****************************************************************************
800 REM     SUBR. ACCEPTOR HANDSHAKE                                               *
810 REM*****************************************************************************
820 ::
830 POKEA5,255:                            REM CLEAR DATA INPUT
840 POKEA4,PEEK(A4)AND(255-8):             REM SET DAC=0
850 POKEA3,PEEK(A3)OR2:                    REM SET RFD=1
860 POKEA3,PEEK(A3)OR4:                    REM SET ATN=0
870 T=PEEK(A3)AND(255-128):IFT=1THEN870:REM WAIT UNTIL DAV=1
880 POKEA3,PEEK(A3)AND(255-2):             REM SET RFD=0
890 B=(255-PEEK(A5)):                      REM B=INPUT DATA
900 POKEA4,PEEK(A4)OR8:                    REM SET DAC=1
910 T=PEEK(A3)OR128:IFT=0THEN910:          REM WAIT UNTIL DAV=0
920 POKEA4,PEEK(A4)AND(255-8):             REM SET DAC=0
930 POKEA3,PEEK(A3)OR2:                    REM SET RFD=1
940 RETURN
950 ::
960 ::
970 REM*****************************************************************************
980 REM     SUBR. READ ERROR                                                       *
990 REM*****************************************************************************
1000 ::
1010 T=B AND 15:                           REM MASK BIT 4-7 OF STATUS BYTE
1020 IFT=1THENC$="OL":                     REM OVERLOAD
1030 IF T=2THENC$="CF":                    REM CRESTFACTOR EXCEEDED
1040 IF T=3THENC$="OL+CF":                 REM OVERLOAD+CRESTFACTOR EXCEEDED
1050 RETURN
```

2.2.6. Simple program examples.

It is not always necessary to use the previously mentioned subroutines especially where the program execution time, memory space or the time to load the program are the main consideration. The program should be as short as possible.
Some simple program examples without the use of the Service Request function are shown below:

2.2.6.1. Program example to control the PM6674 counter.

```
100 GOTO 160
110 ***********************************
120 *                                 *
130 *   PM6674 <IEC> CBM COMMODORE    *
140 *                                 *
150 ***********************************
160 PRINT CHR$(147)                        :REM CLEAR SCREEN.
170 OPEN3,10                               :REM ASSIGN COUNTER TO FILE 3.
180 INPUT"FUNCTION: ";FF$
190 PRINT CHR$(147)                        :REM CLEAR SCREEN.
200 PRINT"FUNCTION CHOOSEN: ";FF$
210 :
220 PRINT#3,FF$;"T1"                       :REM DRIVES THE COUNTER INTO
230 :                                      :REM   THE SELECTED FUNCTION (FF$)
240 :                                      :REM   AND SETS DISPLAY HOLD.
250 PRINT#3,"P"                            :REM DRIVES THE COUNTER TO SEND
260 :                                      :REM   INFORMATION ABOUT THE
270 :                                      :REM   PROGRAMMED FUNCTIONS ETC.
280 PRINT"PROGRAM: ";
290 INPUT#3,A$,B$,C$,D$,E$,F$,G$           :REM READS PROGRAM ITEMS FROM
300 :                                      :REM   COUNTER.
310 PRINTA$;B$;C$;D$;E$;F$;G$              :REM DISPLAYS PROGRAM ITEMS.
320 :
330 FOR I=1 TO 20                          :REM LOOP FOR 20 MEASUREMENTS.
340 :
350 : :PRINT#3,"X"                         :REM TRIGGERS COUNTER TO START
360 :                                      :REM   THE MEASUREMENT.
370 : :INPUT#3,R$                          :REM READS MEASURED RESULT.
380 : :IF ST=2 THEN 500                    :REM IF A TIME-OUT OCCURS
390 :                                      :REM   EXECUTE FIRST OTHER PART.
400 : :PRINT R$                            :REM DISPLAYS MEASURED RESULT.
410 :
420 NEXT I
430 GOTO 180                               :REM REPEATS PROGRAM FOR AN
440 :                                      :REM   OTHER FUNCTION.
450 CLOSE3                                 :REM CLOSES THE ASSIGNED FILE.
460 END
500 PRINT" OTHER PART OF PROGRAM"
510 FOR P=1 TO 1500:NEXT P
520 GOTO 370
READY.
```

2.2.6.2. Program example to control the PM2528 multimeter.

```
100 GOTO 160
110 ********************************
120 *                              *
130 *   PM2528 <IEC> CBM COMMODORE   *
140 *                              *
150 ********************************
160 OPEN3,22                          :REM ASSIGN PM2528 TO FILE 3.
170 PRINT#3,"F00,R0,T1,D1"            :REM VDC,AUTO-RANGE,EXT. TRIGGER,
180 :                                 :REM  D1 PREVENTS ASYNCHRONOUS
190 :                                 :REM  INTERRUPT IN STATEMENT 210.
200 PRINT#3,"E1"                      :REM START MEASUREMENT.
210 INPUT#3,A$:IF ST=2 THEN 210       :REM READ MEASUREMENT DATA ;WHEN NOT
220 :                                 :REM  AVAILABLE WITHIN 65 MSEC (ST=2)
230 :                                 :REM  THEN INTERRUPT HANDSHAKE AND
240 :                                 :REM  REPEAT INPUT.
250 PRINT A$                          :REM DISPLAY MEASUREMENT DATA.
260 GOTO 200                          :REM REPEAT MEASUREMENT.
READY.
```

Ordering number 9499 997 00111

Printed in the Netherlands — 811201